Class Visualization of High-Dimensional Data with Applications

Inderjit S. Dhillon, Dharmendra S. Modha, and W. Scott Spangler
IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120-6099
Email for correspondence: dmodha@almaden.ibm.com

August 10, 1999

Abstract

We consider the problem of visualizing high-dimensional data that has been categorized into various classes. Our goal in visualizing is to quickly absorb inter-class and intra-class relationships.

Towards this end, we introduce class-preserving projections of the multidimensional data onto two-dimensional planes which can then be displayed on a computer screen. These class-preserving projections maintain the high-dimensional class structure, and are closely related to Fisher’s linear discriminants. By displaying sequences of such two-dimensional projections and by moving continuously from one projection to the next, we can create illusions of smooth motion through a multidimensional display. We call such sequences class tours. Furthermore, we overlay class-similarity graphs on our two-dimensional projections to capture the distance relationships in the original high-dimensional space.

We illustrate the above visualization tools on the classical Iris plant data, the ISOLET spoken letter data, and the PENDIGITS on-line handwriting data set. We show how our visual examination of the data can uncover latent class relationships.

Keywords: class-preserving projections, classification, class tours, clustering, linear projections, motion graphics, multidimensional visualization, similarity graphs.
1 Introduction

Classification and clustering are central tools in machine learning and data mining, and are used in a variety of applications such as fraud detection, signal processing, time-series analysis and optical character recognition. The Yahoo! hierarchy of the World-Wide Web is a prime example of the value of classification (www.yahoo.com). However, as recognized in [1, p.269]:

"The use of any clustering method necessarily confronts the user with a set of “clusters” whether or not such clusters are meaningful. Thus, from a data-analytic viewpoint, there is a crucial need for procedures that facilitate the interpretation of the results and enable a sift of useful findings from less important ones and even methodological artifacts. Formal tests for the statistical significance of clusters have been proposed but informal more data-oriented methods that make fewer assumptions are also needed".

Visualization is one such effective and informal procedure. In this paper we propose a scheme for visually understanding inter-class and intra-class relationships.

We will assume that the data is embedded in high-dimensional Euclidean space $\mathbb{R}^d$, and that proximity in $\mathbb{R}^d$ implies similarity. The data may naturally occur in this form, or vector-space models of the underlying data may be constructed, for example, voice, images or text documents may be treated as vectors in a multidimensional feature space, see [2, 3, 4, 5]. We will also not worry about how the data is classified — the classification may be done manually as in the Yahoo! hierarchy, or may be obtained by clustering methods such as the $k$-means or vector quantization algorithms [6, 7, 8]. We assume that we know the representation of all the data points as vectors in $\mathbb{R}^d$, and their respective class labels.

Our main aim in this paper is to visually understand the spatial relationships between various classes in order to answer questions such as:

1. how well-separated are different classes?
2. what classes are similar or dissimilar to each other?
3. what kind of surface separates various classes, for example, are the classes linearly separable?
4. how coherent or well-formed is a given class?

Answers to these questions can enable the data analyst to infer inter-class relationships that may not be part of the given classification, and additionally, gauge the quality of the classification and quality of the feature space. Discovery of interesting class relationships in such a visual examination can help in the design of a better classifier and also lead to enhanced feature selection. We can term this visual process as visual discriminant analysis. Such an analysis would be useful while training a classifier in a “pre-classification” phase, or in evaluating the quality of clusters in a “post-clustering” phase.

Towards this end, we propose the use of carefully chosen two-dimensional projections to display the data. In this paper, our main contributions are:

Class-preserving projections as the main tool for visual explorations. Class-preserving projections are linear projections of the data onto two-dimensional planes that attempt to
preserve the inter-class structure present in the original multidimensional space $\mathbb{R}^d$. The underlying theory, supported by several illustrations on the Iris plant and ISOLET speech recognition data sets [9], is presented in Section 2.

**Class-similarity graphs** to enhance each individual two-dimensional projection of the data. Such graphs provide a skeleton of the data and serve as guides through the various projections (see Section 3).

**Class tours** that show sequences of the two-dimensional class-preserving projections to create the illusion of smooth motion through a multidimensional display (see Section 4). Class tours allow us to "view" higher-dimensional subspaces.

We illustrate the value of the above visualization tools in Section 5, where we present a detailed study of the PENDIGITS on-line handwriting recognition data set [9]. This visual examination allows us to uncover some surprising class relationships.

Projecting high-dimensional data to lower dimensions is an old trick that has been profitably exploited in visualization and other applications, see, for example, principal components analysis [6], projection pursuit [10, 11], Kohonen’s self-organizing maps [12] and multidimensional scaling [13, 14]. For visualization purposes, two-dimensional projections based on principal components, canonical correlations and data sphering were considered in [15]. Later, [16, 17] also incorporated projections based on projection pursuit. The non-linear Kohonen self-organizing maps were used for visualization in [18, 19, 20].

Our class-preserving projections are tailored for exposing the class structure of multidimensional data, and are most closely related to the projections of classical linear discriminant analysis [6], see the end of Section 2.1 for a comparison. Our visualization scheme is closest to the work of Gnanadesikan, Kettenring, and Landwehr [1]. These authors considered two types of projections: (a) local projections that focus on one class and show its relationship to nearby classes, and (b) a global projection onto the two leading principal components of the between-class scatter matrix, $S_B$. The latter projection is very similar to one of our class-preserving projections, see Section 2.2.1 for details. However our work differs from [1] in many ways. We have considered a much wider variety of projections, each of which gives a local view of a user-specified subset of the classes. We also augment our projections with class-similarity graphs and attempt to view projections onto the entire column subspace of $S_B$, and not just onto the two leading principal components. We achieve the latter goal by showing sequences of two-dimensional projections, thus simulating a multidimensional display. We call such sequences class tours.

Other types of tours have been considered earlier. Asimov, Buja and colleagues first proposed grand tours in [21, 22, 23]. A grand tour displays a carefully designed sequence of two-dimensional projections that cover the entire high-dimensional space. Grand tours are independent of the underlying data distribution. However, when the dimensionality of the original data is high, these tours become less informative and computationally infeasible. This problem was alleviated by using data-dependent guided tours which show sequences of projections using principal components, projection pursuit, etc. [15, 16, 17]. The interactive dynamic data visualization software, XGobi [24] contains these guided tours. For a more detailed review of visualization schemes for high-dimensional data, see [25].

Based on the ideas in this paper, we have implemented a software tool named CViz. The CViz software is written in the platform-independent JAVA language, and is currently available
as free test software from IBM's Alphaworks site, www.alphaworks.ibm.com/formula/cviz. In fact, most of the figures in this paper are screen shots of plots produced by CViz.

An earlier, shorter version of this paper, where the focus was on visualizing high-dimensional data arising from text documents, was presented at the 1998 Interface Conference [26]. We have also successfully used our projection scheme in an application other than visualization; that of constructing compact representations of large, sparse text data arising in text mining and information retrieval [27].

2 Class-Preserving Projections

Our main tool for visualizing multidimensional data will be linear projections onto 2-dimensional planes which can then be displayed on a computer screen. The key difficulty is the inevitable loss of information when projecting data from high-dimensions to just 2 dimensions. This loss can be mitigated by carefully choosing the 2-dimensional planes of projection. Thus, the challenge is in the choice of these planes. We want to choose those planes (projections) that best preserve inter-class distances.

2.1 Discriminating Three Classes

We first consider the canonical case where the data is divided into three classes. Let $x_1, x_2, \ldots, x_n$ denote the $d$-dimensional data points divided into the three classes $C_1, C_2$ and $C_3$. The corresponding class-means (or class-centroids) are defined as

$$m_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i, \quad j = 1, 2, 3,$$

where $n_j$ is the number of data points in $C_j$.

For the purpose of visualization, we want to linearly project each $x_i$ onto a 2-dimensional plane. Let $w_1, w_2 \in \mathbb{R}^d$ be an orthonormal basis of the candidate plane of projection. The point $x_i$ gets projected to the pair $(w_1^T x_i, w_2^T x_i)$ and consequently, the means $m_j$ get mapped to

$$(w_1^T m_j, w_2^T m_j), \quad j = 1, 2, 3.$$

One way to maintain good separation of the projected classes is to maximize the distance between the projected means. This may be achieved by choosing vectors $w_1, w_2 \in \mathbb{R}^d$ such that the objective function

$$Q(w_1, w_2) = \sum_{i=1}^{3} \left( |w_1^T (m_2 - m_1)|^2 + |w_2^T (m_3 - m_1)|^2 + |w_2^T (m_3 - m_2)|^2 \right)$$

$^1$there have been over 2000 downloads of the CViz software since it was first released in June, 1998
Figure 1: A Class-Preserving Projection of the Iris data

is maximized. The above may be rewritten as

\[
Q(w_1, w_2) = \sum_{i=1}^{2} \left\{ w_i^T \{(m_2 - m_1)(m_2 - m_1)^T + (m_3 - m_1)(m_3 - m_1)^T \\
+ (m_3 - m_2)(m_3 - m_2)^T\}w_i \right\}
\]

\[
= w_1^T S_B w_1 + w_2^T S_B w_2
\]

\[
= \text{trace}(W^T S_B W),
\]

where

\[
W = [w_1, w_2], \quad w_i^T w_2 = 0, \quad w_i^T w_i = 1, \quad i = 1, 2, \quad \text{and}
\]

\[
S_B = (m_2 - m_1)(m_2 - m_1)^T + (m_3 - m_1)(m_3 - m_1)^T + (m_3 - m_2)(m_3 - m_2)^T. \quad (1)
\]

The positive semi-definite matrix $S_B$ can be interpreted as the inter-class or between-class scatter matrix. Note that $S_B$ has rank \leq 2, since $m_3 - m_2 \in \text{span}[m_2 - m_1, m_3 - m_1]$.

It is clear that the search for the maximizing $w_1$ and $w_2$ can be restricted to the column (or row) space of $S_B$. But as we noted above, this space is at most of dimension 2. Thus, in general, the optimal $w_1$ and $w_2$ must form an orthonormal basis spanning the plane determined by the vectors $m_2 - m_1$ and $m_3 - m_1$. In the degenerate case when $S_B$ is of rank one, that is, when $m_1$, $m_2$ and $m_3$ are collinear, $w_1$ should be in the direction of $m_2 - m_1$ while $w_2$ can be chosen to be any unit vector orthogonal to $w_1$.

Geometrically, the plane spanned by the optimal $w_1$ and $w_2$ is parallel to the plane containing the three class-means $m_1$, $m_2$ and $m_3$. It should be noted that projection onto this plane exactly preserves the distances between the class-means, that is, the distances between the projected means are exactly equal to the corresponding distances in the original d-dimensional space. Thus, in an average sense, we can say that inter-class distances are preserved, and we call such a projection a class-preserving projection.

We illustrate such a class-preserving projection on the famous Iris plant data set in Figure 1. This data set is 4-dimensional and contains 50 members in each of the three classes of plants:
Iris setosa, Iris versicolor and Iris virginica (the 4 dimensions are sepal length, sepal width, petal length and petal width). Note that the three classes are well separated in this figure. For comparison, in Figure 2 we have shown the projection onto a poorly chosen plane where the distinction between the classes is lost. Figure 1 allows us to infer that the Iris setosa plants (on the left part of the figure) are easily distinguished by the feature set from the other two Iris varieties, while the latter two are harder to distinguish. Furthermore, we see that the Iris setosa class is linearly separable from the other classes (since linear separability in any two-dimensional projection implies linear separability in the entire 4-dimensional space).

Projection schemes similar to ours have previously been used in classical linear discriminant analysis. In particular, our class-preserving projections are closely related to Fisher’s linear discriminant and its generalizations. The latter differ slightly from our optimal; they maximize a ratio, such as,

\[
\frac{\text{trace}(W^T S_B W)}{\text{trace}(W^T S_W W)},
\]

where \( S_B \) and \( W \) are as in (1), and \( S_W \) is the within-class scatter matrix, see [6, p.117]. We have chosen to ignore such within-class scatter for computational reasons as we plan to deal with very large and high-dimensional data sets. For more details on linear discriminant analysis, the reader is referred to [28, 6, 29, 30]. Earlier in this section, we observed that our class-preserving projections preserve distances between the three class-means, that is, the multidimensional scaling error for these class-means is zero. For the more general problem of preserving inter-point distances between all the projected data points, see [13, 14, 6].

2.2 Discriminating More than Three Classes

In the above discussion, we arrived at a certain two-dimensional projection that “best” preserves the relationship between three classes. But in most practical situations, we will encounter a greater number of classes. The Yahoo! hierarchy, for instance, segments web pages into thousands of categories. What projections will enable us to examine the interplay between these multitude of classes?
Figure 3: Class-preserving projection discriminates B, C and F in the ISOLET data

Clearly, one option is to examine the k classes three at a time by taking projections identical to those described in the previous section. There are a total of \(^{k \choose 3}\) such 2-dimensional projections, each determined by three class-means. Each of these projections gives us a local view of three classes. We illustrate such local views on an interesting speech recognition data set, that we first describe.

The ISOLET (Isolated Letter Speech Recognition) data consists of speech samples of the 26 English alphabets [2], and is publicly available from the UCI Machine Learning Repository [9]. The data was collected from 150 subjects who spoke the “name” of each letter twice. After some preprocessing, 617 real-valued attributes were chosen to describe each spoken letter. These features consist of various spectral coefficients (sonorant, pre-sonorant, post-sonorant, etc.) of the sampled data, plus other measurements that capture pitch and amplitude characteristics. For a complete list of the feature set and a detailed description, the reader is referred to [2]. Each of the 617 features is mapped to the interval [0.0, 1.0] and is normalized to utilize this entire range. The entire data set comprises 7,797 samples.

For clarity of presentation, we will only consider the samples of the seven spoken letters A through G. There are a total of 2,098 such samples. Note that this data set is extremely high-dimensional (d = 617) and occupies nearly 5 MBytes of memory. In their paper [2], Fanty and Cole designed a neural network classifier for spoken letter recognition. Their classifier used domain knowledge to achieve high accuracy, for example, they trained a separate classifier to distinguish between the letters in the E-set, namely B, C, D and E. We now see how our class-preserving projections enable us to visually uncover such domain knowledge, and capture various *inter-letter* and *intra-letter* relationships in this multidimensional speech data set.

In Figure 3, we show the class-preserving projection determined by the centroids (or means)
Figure 4: Projection to discriminate B, C and F in the ISOLET data — the A, D, E and G data samples are not shown.

of the spoken letters B, C and F. We depict each data point by the corresponding lower-case letter, while the centroid is denoted by the upper-case letter. In Figure 3, the upper-case letters enclosed by a box, that is, \[ B, C \] and \[ F \], denote the centroids that determine the class-preserving projection. We will use this convention throughout the remaining plots in this paper. In our ensuing discussion, we will “overload” the upper-case letters A,B,...,G to denote various quantities. Thus the letter A may indicate (i) the spoken letter A, (ii) the class containing all the data samples of the spoken letter A, or (iii) the corresponding class-centroid. The particular usage should be clear from the context. The origin along with a pair of orthogonal axes is also shown in all our figures.

Figure 3 shows that the classes B, C and F are well-separated and equally far apart. The D and E classes seem to be quite close to B, while most A samples lie between B and F. Since all the 2,098 data points are displayed, Figure 3 looks very crowded. Note that the A, D, E and G samples do not influence the choice of the plane of projection. We can hide such secondary samples in order to get a clearer view of the discriminated classes. For example, Figure 4 shows exactly the same projection as Figure 3 except that A, D, E and G data samples are not displayed. Only centroids of these classes are shown.

Figure 4 shows a projection expressly chosen to preserve only the distances between the B, C and F centroids. Distances from the other centroids are not preserved by this projection. Thus proximity of the D and E centroids to the B centroid in this figure may be misleading, that is, although D and E appear close to B in this 2-dimensional projection they may be quite far in the original 617-dimensional space. To check if B and D are close, we look at the class-preserving projection that preserves distances between B, D and F in Figure 5. Here we see
Figure 5: Projections to discriminate B, D, F and B, E, F in the ISOLET data.

Figure 6: Projection to discriminate A, B and F in the ISOLET data.
that $D$ is indeed quite close to $B$, and they are both well separated from $F$. The right plot in Figure 5 shows a similar relationship between $B$, $E$ and $F$.

In all the above figures we see that $A$ lies between $B$ and $F$. We may suspect that $A$ is closer to $F$ than these figures indicate (note that none of the above figures preserve distances to $A$). However, in Figure 6, the class-preserving projection determined by $A$, $B$ and $F$ shows that $A$ and $F$ are not particularly close. In fact, the $A$ and $F$ classes are seen to be linearly separable. In Figure 6, if we project all the data samples onto the X-axis, we see that $F$ samples are closer to $A$ than to $B$. Thus, the $F$ samples are seen to share some correlated features with $A$ that distinguish them from $B$.

### 2.2.1 Class-Eigenvector Plots

The ISOLET data that we have considered contains seven classes. However, thus far we only have a mechanism to display the inter-class relationships between three classes at a time. In general, we would like to differentiate between more than three classes in the same view.

More formally, we want to obtain a 2-dimensional projection that best discriminates the $q$ classes with class-means $m_1, m_2, \ldots, m_q$, each containing $n_1, n_2, \ldots, n_q$ data points, respectively. Taking an approach similar to that of Section 2.1, we can formulate the above objective as the search for orthonormal $w_1, w_2 \in \mathbb{R}^d$ that maximizes

$$Q(w_1, w_2) = \text{trace}(W^T S_B W),$$

where

$$W = [w_1, w_2], \quad w_1^Tw_2 = 0, \quad w_i^Tw_i = 1, \quad i = 1, 2,$$

and

$$S_B = \sum_{i=2}^{q} \sum_{j=1}^{i-1} n_i n_j (m_i - m_j)(m_i - m_j)^T.$$

Note that the positive semi-definite matrix $S_B$ has rank $\leq (q-1)$ since the vectors $m_i - m_j$, $j \neq 1$ are linearly dependent on the $q-1$ vectors $m_i - m_1$, $i = 2, \ldots, q$. It is well known that the vectors $w_1$ and $w_2$ that maximize the objective function in (2) are the eigenvectors (or principal components) corresponding to the two largest eigenvalues of $S_B$. The reader should note that for $q > 3$, in general, there is no two-dimensional plane that exactly preserves the distances between the $q$ centroids $m_1, m_2, \ldots, m_q$. The plane spanned by the optimal $w_1, w_2$ preserves inter-class distances to the largest extent possible, where the error due to projection is measured in the 2-norm or Frobenius norm, or any unitarily invariant norm [31].

The reader might have noticed the extra factor $n_i n_j$ in (3) that was not present in (1). By weighting each $(m_i - m_j)(m_i - m_j)^T$ term in (3) by the factor $n_i n_j$, we are placing greater emphasis on preserving distances between the class-means of larger classes.

The matrix $S_B$ as given above in (3) is the sum of $\binom{q}{2}$ rank-one matrices. We can show that $S_B$ can be expressed more compactly as the sum of $q$ rank-one matrices. In particular,

$$S_B = n^{(q)} \sum_{i=1}^{q} n_i (m_i - m_i^{(q)})(m_i - m_i^{(q)})^T,$$
Figure 7: Projection to discriminate B, D, E and F in the ISOLET data

where

\[ n^{(q)} = n_1 + n_2 + \cdots + n_q, \quad \text{and} \]
\[ m^{(q)} = \frac{1}{n^{(q)}} (n_1 m_1 + n_2 m_2 + \cdots + n_q m_q) \]

is the mean of all the data points in the q classes under consideration. This alternate expression for \( S_q \) is more compact and better for computational purposes. The interested reader can look at the Appendix for a proof of the equivalence of formulae (3) and (4).

In Figure 7, we show a 2-dimensional projection that preserves inter-class distances between four of the seven classes — B, D, E and F. This view confirms our earlier observations that B, D and E are close to each other and quite distant from F. Figure 8 attempts to differentiate between the B, C, D, E and G classes. Here we see that G is closer to B, D and E while C is more distant. Observe that A and F appear quite close to each other and to B, D and E in Figure 8, but note that this projection does not preserve distances to the A and F classes. Recall Figure 6 which showed that A and F are not so close. Figure 9 has \( q = k = 7 \), and gives a more accurate estimate of the relationships between all the classes.

Finally, in Figure 10, we consider all 7,797 data samples of the 26 English alphabets that comprise the complete ISOLET data. This figure shows a projection that tries to discriminate between all 26 letters. Note that the letters B, C, D, E, G, P, T, V and Z, which constitute the so-called E-set are seen to be very close in this projection.

Thus, given \( k \) classes we have a mechanism for viewing the inter-class relations between any
Figure 8: Projection to discriminate B, C, D, E and G in the ISOLET data.

Figure 9: Projection to discriminate all the 7 letters in the ISOLET data.
Figure 10: Projection to discriminate all 26 letters in the entire ISOLET data

$q$ of them. There are a total of

$$\sum_{q=3}^{k} \binom{k}{q} = 2^k - \frac{k(k+1)}{2} - 1$$

such informative class projections. However, as $q$ gets larger, the distinction between classes starts getting blurred in these views. Figure 10 gives a crowded view that attempts to discriminate all 26 classes. Such crowding is inevitable since our linear projections are limited to 2-dimensional planes. Ideally, we would like to “view” projections onto higher-dimensional subspaces.

2.2.2 Projections to Higher-Dimensional Subspaces

Suppose for a moment that we can visualize projections of the data onto a $p$-dimensional subspace, $p > 3$. As before, we want to preserve distances between the centroids after projection. Thus, given $k$ classes, we want to find orthonormal $w_1, w_2, \ldots, w_p \in \mathbb{R}^d$ such that the objective function

$$Q(w_1, \ldots, w_p) = \text{trace}(W^T S_B W),$$

is maximized, where

$$W = [w_1, \ldots, w_p], \quad w_i^T w_j = 0, \quad w_i^T w_i = 1, \quad 1 \leq i, j \leq k, \quad i \neq j,$$

and $S_B$ is as in (3) or (4) with $q$ replaced by $k$. 

13
The optimal $w_i$ are given by the $p$ eigenvectors (principal components) of $S_B$ corresponding to its $p$ largest eigenvalues. When $p = k - 1$, the desired subspace is the entire column (or row) subspace of $S_B$, and the $w_i$ can be any orthonormal basis of this subspace. In this case, distances between all the class-means are exactly preserved.

Of course, it is not straightforward to visualize a $p$-dimensional subspace, $p > 3$. We are limited to 2-dimensional computer displays. In Section 4, we propose a way to explore the entire column space of $S_B$. But before we do so, we look at a tool that enhances each individual projection.

3 Class-Similarity Graphs

Two-dimensional linear projections are a continuous transformation of the data; two points which are close in $\mathbb{R}^d$ will remain close in each of these projections. However, two points which are close in a 2-dimensional projection need not be close in the original space $\mathbb{R}^d$. To mitigate this information loss, we propose the addition of class-similarity graphs to our 2-dimensional plots.

A class-similarity graph is described as follows [6, p.238]. The vertices of this graph are the class centroids $m_1, m_2, \ldots, m_k$, and there is an edge between the means (or vertices) $m_i$ and $m_j$ if

$$d_2(m_i, m_j) \leq \tau,$$

(5)

where $d_2$ denotes the Euclidean distance, while $\tau$ is a user-controlled threshold parameter. If $\tau$ is very large, then all centroids will be connected. On the other hand, if $\tau$ is very small, then no centroids will be connected. It is thus intuitively clear that the choice of this threshold parameter is important in revealing similarities (or dissimilarities) between the class-means. To obtain “natural” connections between the centroids, $\tau$ will have to be greater than typical distances between related classes but less than typical distances between unrelated ones.

We can display class-similarity graphs by overlaying them on our 2-dimensional class-preserving projections. Figure 11 shows one such graph on the projection that discriminates
Figure 12: B, D and E form a clique in the similarity graph.

Figure 13: B, D, E and G form a clique at a higher threshold $\tau$. 
between all the seven classes (this projection is identical to the one in Figure 9). Note that to show the similarity graph clearly in this figure, we have shown only the seven centroids and removed the individual data points. The B, D and E classes are seen to form a clique, which is consistent with our observations about their closeness in Figures 7, 8 and 9. The class-similarity graph provides a skeleton of the data and reminds us of the proximity relationships between classes through various views of the data. For example, Figure 12 shows the same similarity graph of Figure 11) overlaid on the projection that discriminates between B, D and E. This graph reminds us that B, D and E are the closest among the seven letters even though they appear far apart in this view. Thus, the class similarity graph adds another valuable information dimension to linear projections. Finally, in Figure 13 we display the similarity graph at a higher threshold τ (see (5) above), and observe that B, D, E and G form a clique indicating the nearness of these letters.

In the context of speech recognition systems, the closeness of B, D, E and G to each other indicates that these letters may be the most difficult to classify and hence “recognize”. Indeed, in [2, p.223] the authors remark that they “trained separate (neural) networks for just the letters in the E-set” (see also [32]).

4 Class Tours

Thus far we have limited ourselves to static 2-dimensional photographs of the data set, each of which conveys some limited information. Projection onto the entire (k−1)-dimensional subspace spanned by the vectors mi − m1, i = 2, . . . , k, contains more global inter-class information since it exactly preserves the distances between all the k class-means. Ideally, we would like a mechanism for viewing this inter-class differentiating or class-preserving linear subspace.

In order to simulate multidimensional displays, Asimov proposed the use of motion graphics in [21]. Specifically, he introduced the concept of tours which are sequences of 2-dimensional projections interspersed with a number of intermediate projections. These intermediate projections are obtained by interpolation, and thus tours create an illusion of continuous smooth motion through a multidimensional display. The grand tours proposed in [21, 22] try to display a carefully designed sequence of 2-dimensional projections that are dense in the set of all such projections. However, such sequences are independent of the data set to be visualized and require enormous computational resources when the data is high-dimensional. Guided tours proposed in [15] alleviate this problem by choosing sequences tailored to the underlying data; these tours may be guided by principal components, canonical correlations, data sptering and projection pursuit. See [15, 16, 17] for details.

We have found our class-preserving projections to give good local data displays, for example, see the Figures in Section 2. To get a more global view of the data, we propose class tours which are sequences of our class-preserving 2-dimensional projections, and are an effective tool to “view” the (k−1)-dimensional class-preserving subspace. Metaphorically speaking, a class tour constructs a dynamic, global “movie” of this subspace from a number of static, local photographs.

The basic idea behind class tours is simple: choose a target 2-dimensional projection from the subset of nearly 2^k class-preserving projections, move smoothly from the current projection to this target, and continue. The main questions of interest are (a) the choice of the intermediate 2-dimensional projections, and (b) the choice of the orthonormal basis used for viewing
each projection so that the motion appears smooth. [21, 22] have proposed using geodesic interpolation paths between the current and the target planes. Each geodesic path is simply a rotation in the (at most) 4-dimensional linear subspace containing both the current and the target 2-d planes. Various smoothness properties of such geodesic paths are explored in great detail in [23].

A geodesic path between a current plane, \( U_0 \), and a target plane, \( U_1 \) may be constructed as follows.

1. Compute the so-called principal vectors \( x_0, x_1, y_0, y_1 \), and associated principal angles \( \theta, \phi \) by the following procedure. Note that the vectors \( x_0 \) and \( x_1 \) should satisfy the important property that they make the smallest possible angle amongst all pairs of vectors drawn from \( U_0 \) and \( U_1 \).

   (a) Given arbitrary orthonormal basis vectors, \( (x_0, y_0) \in U_0 \) and \( (x_1, y_1) \in U_1 \), compute \( \delta_0, \delta_1 \) such that the dot product

   \[
   (x_0 \cos \delta_0 + y_0 \sin \delta_0, x_1 \cos \delta_1 + y_1 \sin \delta_1)
   \]

   is maximized. Then, for the maximizing \( \delta_0, \delta_1 \), the first principal angle and pair of principal vectors are given by

   \[
   x_0 = x_0 \cos \delta_0 + y_0 \sin \delta_0, \quad x_1 = x_1 \cos \delta_1 + y_1 \sin \delta_1, \quad \cos \theta = x_0^T x_1.
   \]

   (b) Orthogonalize \( x_0 \) or \( y_0 \) against \( x_0 \) to obtain \( y_0 \). Similarly obtain \( y_1 \) so that \( x_1 \) and \( y_1 \) form an orthonormal basis for \( U_1 \). Then \( \cos \phi = y_0^T y_1 \).

   For a more general treatment on computing principal angles and vectors, see [33].

2. Orthogonalize \( x_1 \) against \( x_0 \) to obtain the unit vector \( x_1^* \). Similarly, obtain \( y_1^* \) from \( y_0 \) and \( y_1 \).

3. Construct the moving vectors \((x(t), y(t))\) where

   \[
   x(t) = \cos(t \bar{\theta}) x_0 + \sin(t \bar{\theta}) x_1^*, \quad y(t) = \cos(t \bar{\phi}) y_0 + \sin(t \bar{\phi}) y_1^*,
   \]

   where \( \bar{\theta} = \theta / \sqrt{\bar{\theta}^2 + \bar{\phi}^2} \) and \( \bar{\phi} = \phi / \sqrt{\bar{\theta}^2 + \bar{\phi}^2} \). The parameter \( t \) is varied from 0 to \( \sqrt{\bar{\theta}^2 + \bar{\phi}^2} \), and the planes spanned by the bases \((x(t), y(t))\) give a geodesic from \( U_0 \) to \( U_1 \).

Note that the above procedure specifies a geodesic between a current plane and a target plane. However, such a path may be embedded in a long sequence of 2-dimensional planes. In such a case, we must use properly rotated principal vectors for horizontal and vertical projections. This avoids subjecting the user to meaningless within-screen rotations whenever a new geodesic path is to be resumed. Thus the above procedure results in interpolating planes rather than specific pairs of basis vectors. More computational details may be found in [15, section 2.2.1].

Although we have presented the main ideas above, it is hard to illustrate class tours through static photographs. We encourage the interested reader to experiment with our CViz software, which is currently available at www.alphaWorks.ibm.com/forum/cviz.
5 A Case Study

In this section, we present a case study that illustrates how we can use our CViz visualization software to explore a real-life multidimensional handwriting recognition data set. In our visual study, we use all the tools developed in the earlier sections, namely, class-preserving projections, similarity graphs and class tours. As we shall see, our visual exploration allows us to uncover some surprising characteristics of the data set. Although we cannot hope to fully convey the discovery process, we present snapshots of the interesting findings to illustrate how we can incrementally build upon our discoveries.

5.1 The On-line Handwriting Recognition Data Set

The PENDIGITS data set consists of 250 handwriting samples from 44 writers [34, 3], and is publicly available from the UCI Machine Learning Repository [9]. The authors of [3] collected the raw data from a pressure sensitive tablet that sent x and y coordinates of the pen at fixed time intervals of 100 milliseconds. Two preprocessing steps were performed to reduce meaningless variability arising from variations in writing speed and sizes of the written digits:

Normalization makes the data invariant to translations and scale distortions. Both x and y coordinates of the raw data were scaled so that the coordinate which has the maximum range varied between 0 and 100.

Resampling represents each digit as a constant length feature vector. In this data set, spatial resampling with simple linear interpolation was used to obtain 8 regularly spaced points on the trajectory of each digit.

Thus, each digit is represented by 8 (x, y) coordinates leading to a 16-dimensional feature vector. Each of the 16 attributes is an integer varying from 0 to 100. In Figure 14, we show some of the reconstructed digit samples, where the directions of the arrows indicate the pen’s trajectory. The starting point of the pen’s path is indicated by a ‘o’, the end point by a ‘O’, while the 6 intermediate points are marked by ‘+’. Note that these displayed samples are not replicas of the pen’s original trajectory but are reconstructed from the 8 (x, y) coordinates by connecting adjacent coordinates by a straight line. Hence, even though the 8 (x, y) coordinates are regularly spaced in arc length of the pen trajectory, they do not appear so in Figure 14. Also note that since the pen pressure values are not stored the “pen lifts” are lost, as in the handwritten 7 in Figure 14.

The entire PENDIGITS data set consists of 10,992 samples. More information on this data set may be obtained from [34, 3, 9].

5.2 Visual Exploration of the PENDIGITS Data

We start our visual examination with Figure 15 which displays the class-preserving projection that preserves distances between the centroids of digits 0, 1 and 2. As in most of the figures of Section 2.2, we only show centroids of the other classes, and not their individual data points. The centroids that determine the class-preserving projection are denoted by the corresponding digit enclosed by a box, for example, [0], [1] and [2] in Figure 15.

We now enumerate our findings in an order in which the discovery process might progress.
Figure 14: Sample "reconstructed" handwritten digits — 1, 2, 6 and 7
Figure 15: Class-Preserving Projection discriminates between 0, 1 and 2 in the PENDIGITS data

1. Figure 15 shows that the digits 1 and 2 are closer to each other than to 0. The 0 class appears to have a large variance, while the cluster of 2's is more coherent. In fact, we see that the 0 class forms a boomerang-like shape. The large variance implies that there are a variety of ways of writing 0, while the boomerang shape suggests a continuum in the different written 0's. We can delve deeper into this hypothesis by looking at different written 0 samples, some of which are shown in Figure 16. Indeed, we see that these different 0's seem to be rotations of each other — each 0 traces an anti-clockwise arc but has different starting and ending points.

In Figure 15, we also see that centroids for the digits 5, 6 and 8 lie near each other, while centroids of 3 and 9 are also close. But this projection tries to preserve distances only between the 0, 1 and 2 centroids, and the proximity of other digits may be misleading here. To get an estimate of inter-class distances in the original 16-dimensional space, we turn to class-similarity graphs.

2. A class-similarity graph, at an interesting threshold \( \tau \), is shown in Figure 17. Note that this projection is identical to the earlier one in Figure 15. Since they are connected by edges, centroids for digits 2 & 7, and digits 3 & 9 are close to each other in the original space. The similarity between the written digits 2 and 7 is also clearly seen in Figure 14. The third “edge” in the class-similarity graph seems to connect the three digits 5, 6 and 8. However it is not clear if this triplet forms a clique, so we look at the same class-similarity graph in another projection.

3. Figure 18 clearly shows that the digits 5 and 8 that are close. However, 6 is not particularly
Figure 16: Different handwritten zeros

Figure 17: Class similarity graph for the PENDIGITS data
close to either 5 or 8. Note that the projection in Figure 18 attempts to discriminate between all ten digits and gives a good estimate of their relative distances from each other.

4. We may now wish to examine the pairs 2 & 7, 3 & 9, 5 & 8 more closely. To do so, we look at each pair relative to the 0 class. These three projections are shown in Figures 19, 20 and 21. These views allow us to make the following inferences:

(a) The two pairs 2 & 7, 3 & 9 appear quite close to each other, and rather distant from 0.

(b) Figure 21 shows that some of the 8's are similar to the 0's. Some other samples, such as 5's and 8's, also appear intermingled. In the context of designing a classifier, these samples are liable to be misclassified.

(c) The classes 2 and 3 are very coherent and indicate a consistency among the writers in writing these digits.

(d) The classes 8 and 9 are seen to have large variances. This observation is confirmed by the variety of ways of writing 8 and 9 seen in Figures 22 and 23.

5. While surfing through other projections, we found a particularly interesting one that we now present. Figure 24 attempts to maximally preserve the distances between the centroids of digits 3, 5, 8 and 9. As expected, we see that 3 and 9 samples lie close to each other. However, the class-structure of 5 is intriguing. The 5 samples appear in two very distinct clouds — the first is near 3 and 9, while the second is near 8 but far from 3 and 9. The centroid for the digit 5 is seen to lie in the middle of these two clouds and very close to the 8 centroid, but there are no individual data samples near the centroid! This strange behavior invites a closer look at handwritten samples of 5.

Figure 25 shows two quite different ways of writing 5 and explains the two different clouds in Figure 24. The left plot of Figure 25 gives the cursive way of writing 5 where
Figure 19: Projection to discriminate 0, 2 and 7

Figure 20: Projection to discriminate 0, 3 and 9
Figure 21: Projection to discriminate 0, 5 and 8

Figure 22: Different types of 8
Figure 23: Different types of 9

Figure 24: Projection to discriminate 3, 5, 8 and 9
there is no “pen lift”. Such 5’s lie in the cloud near 9 and 3. Indeed, the left plots in Figures 23 and 25 exhibit the similarity between 9 and the cursive 5. The right plot of Figure 25 has the non-cursive 5 where the writer lifts the pen during writing. These 5’s are somewhat similar to some of the 8’s shown in Figure 22. The two clouds of 5 are also visible in Figure 26, which displays a projection that attempts to discriminate between all 10 digits.

This visual discovery is rather surprising and can help in the design of a better handwriting recognition system by making the classifier recognize two types of 5, and thus recognize eleven “different” digits.

6. Finally, in Figure 27 we draw the centroids of all the ten digits in the same manner we displayed individual digit samples in Figure 14. We observe that the class centroids for 2, 3, 4 and 6 look like “normal” written digits. This observation is consistent with our visual explorations where we found these classes to be quite coherent, for example, see Figures 19 and 20. We also observe that the 5 and 9 centroids bear little resemblance to the corresponding written digits. The centroid for 5 is the average of the cursive and non-cursive 5, while the average 9 reflects the confusion between the clockwise and anti-clockwise arcs in writing 9.

6 Conclusions

In this paper, we have proposed the use of class-preserving projections for visual discriminant analysis. These projections satisfy a certain optimality criterion that attempts to preserve distances between the class-means. Our projections are similar to Fisher’s linear discriminants that are commonly used in classical discriminant analysis. Class-similarity graphs remind us of proximity relations between classes in each of our class-preserving projections. We also use class tours which enable us to view sequences of class-preserving projections, interspersed with interpolating projections that create an illusion of smooth motion through a multidimensional subspace. Class tours allow the touring of all the data points in an informative inter-class differentiating linear subspace. All the above ideas comprise our visualization software toolbox that allows us to capture the inter-class structure of complex, multidimensional data.
We have illustrated the use of our visualization tool in discovering interesting class relationships in the Iris, ISOLET and PENDIGITS data. Such discoveries underscore the value of visualization as a quick and intuitive way of understanding a data set. For illustration purposes, we have deliberately used intuitive speech and handwriting data sets, where our visual explorations lead to "natural" conclusions. In cases where there is not much domain knowledge about the data, such visual discoveries would be invaluable.

Acknowledgements. We are grateful to Werner Stuetzle for bringing the work in [1] to our attention, and to Sriganes Madhavanath for help in the area of handwriting recognition.

Appendix

Proposition. The matrices $S_B$ given in equations (3) and (4) are identical.
Figure 27: The "average digits"
Proof. Expanding each term in equation (3) and using symmetry in i and j, we get

\[ S_B = \sum_{i=1}^{q} \sum_{j=1, j \neq i}^{q} n_i n_j (m_i m_j^T - m_i m_i^T) \]
\[ = \sum_{i=1}^{q} \left\{ n_i (n_1 + n_2 + \ldots + n_q) m_i m_i^T - n_i^2 m_i m_i^T \right\} \]
\[ - \sum_{i=1}^{q} \sum_{j=1, j \neq i}^{q} n_i n_j m_i m_j^T, \]
\[ = n^{(q)} \sum_{i=1}^{q} n_i m_i m_i^T - \sum_{i=1}^{q} \sum_{j=1}^{q} n_i n_j m_i m_j^T, \]
\[ = n^{(q)} \left( \sum_{i=1}^{q} n_i m_i m_i^T - n^{(q)} m^{(q)} m^{(q)T} \right), \]
\[ = n^{(q)} \sum_{i=1}^{q} n_i (m_i - m^{(q)}) (m_i - m^{(q)})^T. \]

References


